

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

Zadání bakalářské práce

Student:

Daniel Mrózek

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

**Absolvování individuální odborné praxe
Individual Professional Practice in the Company**

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: AUTEL, a.s.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Mgr. Jiří Dvorský, Ph.D.**

Konzultant bakalářské práce: Ing. Tomáš Bajger

Datum zadání: 01.09.2015

Datum odevzdání: 29.04.2016



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 29. dubna 2016


.....

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

AUTEL, a. s.
Oldřichovice 790
739 61 Třinec 1

V Třinci - Oldřichovicích 22. dubna 2016



.....

Rád bych na tomto místě poděkoval firmě Autel, a.s., která mi umožnila vykonávání bakalářské práce formou odborné praxe. Zejména svému konzultantovi, panu Ing. Tomáši Bajgerovi za vedení a čas strávený v průběhu vykonávání praxe a také panu Ing. Richardu Konderlovi. Dále bych chtěl poděkovat doc. Mgr. Jiřímu Dvorskému, Ph.D za pomoc při vypracování této práce.

Abstrakt

Obsahem této bakalářské práce je popis mé pracovní činnosti ve firmě Autel, a. s., kde jsem vykonával odbornou praxi. Nejprve zde informuji o zaměření firmy a pracovní pozici, na kterou jsem byl zařazen. Dále zde uvádím hlavní úkoly přidělené v průběhu praxe, společně s jejich řešením a popisem technologií, se kterými jsem pracoval a používal je. V samotném závěru pak zhodnocuji mé působení ve firmě a celkový přínos absolvované praxe.

Klíčová slova: bakalářská práce, odborná praxe, .NET Framework, EPLAN, CAD/CAE systém, C#, WPF, SQL

Abstract

This thesis contents a description of my work in the Autel, a. s. company, where I did my professional practice. First chapter informs about specialization of the company, and the job, I was assigned to. In further paragraphs I mention the main tasks I was assigned to, during course of practice, together with their solutions and description of technologies I used and worked with. In the conclusion I've summarized my career in the company and the overall benefits of the completed practice.

Key Words: bachelor thesis, professional practice, .NET Framework, EPLAN, CAD/CAE system, C#, WPF, SQL

Obsah

Seznam použitých zkratk a symbolů	8
Seznam tabulek	9
Seznam výpisů zdrojového kódu	10
1 Úvod	11
2 Popis odborného zaměření firmy a pracovní zařazení	12
2.1 Odborné zaměření firmy	12
2.2 Pracovní zařazení	12
3 Seznam zadaných úkolů	13
3.1 Překreslení obvodových schémat	13
3.2 EPLAN skripty	13
3.3 WPF aplikace pro generování a aktualizaci projektů	13
3.4 Napojení aplikace na místní IS	14
4 Zvolený způsob řešení zadaných úkolů	15
4.1 Překreslení obvodových schémat	15
4.2 EPLAN skripty	16
4.3 WPF aplikace pro generování a aktualizaci projektů	19
4.4 Napojení aplikace na místní IS	22
5 Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe	25
6 Znalosti či dovednosti scházející studentovi v průběhu odborné praxe	26
7 Dosažené výsledky v průběhu odborné praxe a její zhodnocení	27
Literatura	28
8 Přílohy	29

Seznam použitých zkratk a symbolů

API	– Application Programming Interface
ASP.NET	– Active Server Pages .NET
ASŘ	– Automatizovaný systém řízení
CAD	– Computer-aided design
CAE	– Computer-aided engineering
CRUD	– Create, Read, Update, Delete
DCS	– Distributed Control System
ERP	– Enterprise Resource Planning
IS	– Informační systém
MES	– Manufacturng Execution Systems
MaR	– Měření a regulace
PLC	– Programmable Logic Controller
SQL	– Structured Query Language
WPF	– Windows Presentation Foundation
XAML	– Extensible Application Markup Language
XML	– Extensible Markup Language

Seznam tabulek

1	Akce XEsSetProjectPropertyAction	18
---	--	----

Seznam výpisů zdrojového kódu

1	Použití akce XEsSetProjectPropertyAction	18
2	Ukázka spuštění skriptu v příkazovém řádku	18
3	Funkce pro převedení na Unixový čas	21

1 Úvod

Při výběru tématu bakalářské práce jsem zjistil, že existuje možnost vykonávat odbornou praxi ve vybrané firmě, proto jsem neváhal a vydal se touto cestou. Rád se učím novým technologiím a řeším problémy vznikající při tvorbě aplikací či menších informačních systémů. Věřil jsem, že tyto znalosti mohu uplatnit a dále rozvíjet ve firmě, a proto jsem si zvolil bakalářskou praxi.

Tato bakalářská práce popisuje mou činnost ve firmě Autel, a. s. V této firmě jsem působil již dříve, ale pouze na dvoutýdenní praxi jako praktikant. Zaujalo mě ale zaměření a celkový chod firmy, a proto jsem se rozhodl právě pro ni. V bakalářské práci budu popisovat jeden hlavní úkol, kterým jsem byl pověřen. Tato firma se mimo jiné zabývá projektováním elektrických instalací ve specializovaném CAD/CAE systému. Vytváření nových projektů v tomto systému a následné nastavování různých vlastností a atributů je zdlouhavá a neefektivní práce. Mým hlavním úkolem tedy bylo zrychlit a zefektivnit tento proces. Tento, z počátku menší úkol, se dále rozrůstal o další funkcionalitu, a proto jsem se rozhodl ho rozdělit na menší úlohy, které budu v této práci postupně popisovat.

V práci nejprve představím firmu, ve které jsem odbornou praxi vykonával a nastíním, čím se zabývá. Stručně také popíšu mou roli ve firmě. U jednotlivých úkolů nejprve uvedu zadání a následně popíšu jejich problematiku a zvolené řešení.

Na konci práce je pak zhodnocení mého působení ve firmě a také mé pracovní náplně. Hodnotím také využití mých dosavadních znalostí nabytých v průběhu studia na vysoké škole a jejich využití při vykonávání odborné praxe. Na závěr shrnu znalosti a zkušenosti nabyté při této praxi.

2 Popis odborného zaměření firmy a pracovní zařazení

2.1 Odborné zaměření firmy

Autel, a. s. [1] (dále jen Autel) je inženýrsko-dodavatelská společnost, která zajišťuje spolehlivé a pružné služby a dodávky nejmodernějších technologií na klíč pro průmyslové celky v těchto oblastech:

- **Projektování.** Skupina projektantů vypracovává projektové dokumentace všech stupňů, zejména pak realizační dokumentace. Projektování se dělí na silnoproudé a MaR a ASŘ.
- **Programování základní úrovně.** Programování PLC a DCS systémů základní úrovně. Vizualizace technologických procesů.
- **Programování nadřazené úrovně.** Vývoj software na zakázku na platformě Microsoft .NET.
- **Inženýrské činnosti.** Komplexní realizace projektu. Vedení projektu, zajišťování dodávek a služeb, uvádění do provozu.
- **MES systémy.** Rozvrhování a řízení výroby. Energetické informační systémy.

2.2 Pracovní zařazení

Ve firmě jsem byl zařazen jako .NET programátor. Má pracovní náplň se ale týkala oddělení projekce. Vyvíjel jsem zde aplikaci pro tvorbu projektu v CAD/CAE systému EPLAN. Tuto aplikaci jsem od začátku až do konce samostatně navrhoval, vyvíjel, implementoval a testoval pod vedením mého konzultanta, který mi také zadával požadavky na tuto aplikaci. Při napojení této aplikace na místní informační systém jsem určitou dobu spolupracoval s pracovníkem IT oddělení.

3 Seznam zadaných úkolů

V této kapitole jsou vypsané úlohy řešené v průběhu odborné praxe. Úlohy jsou seřazeny v pořadí, v jakém byly zadány a vykonány.

Komunikace SQL databáze s CAD/CAE systémem EPLAN

Jedná se o hlavní úkol řešený po celou dobu působení ve firmě. Cílem tohoto úkolu bylo vytvořit aplikaci, která dokáže získat informace o jednotlivých zakázkách z SQL databáze a vytvořit nový projekt v CAD/CAE systému EPLAN (dále jen aplikace EPLAN nebo jenom EPLAN) se získanými informacemi. Pro snadnější realizaci byl úkol rozdělen na menší, snadněji řešitelné úlohy. Jednotlivé části byly postupně odevzdávány a kontrolovány konzultantem a případně rozšiřovány o další funkce.

Cílem tohoto úkolu nebylo vytvořit kompletní komunikaci mezi databází a EPLANEM

3.1 Překreslení obvodových schémat

Úkolem bylo seznámit se s prostředím aplikace EPLAN. V rámci seznamování jsem překresloval starou papírovou dokumentaci do elektronické podoby.

Časová náročnost: Celkově jsem nad tímto úkolem strávil cca 8 dní. Aplikace EPLAN je poměrně rozsáhlá, a proto mi šlo překreslování schémat z počátku trochu pomaleji.

3.2 EPLAN skripty

V dalším úkolu jsem se věnoval EPLAN knihovnám, které slouží pro tvorbu skriptů a jsou poskytovány v základní instalaci aplikace EPLAN. Dále jsem tvořil skript, který změní vlastnosti projektu.

Časová náročnost: Tento úkol mi zabral přibližně 5 dní. Časově nejnáročnější bylo najít řešení, jak v aplikaci EPLAN vytvořit nový projekt bez použití uživatelského rozhraní aplikace. Po nalezení řešení již trvalo pouze pár hodin vytvořit daný skript a otestovat jej.

3.3 WPF aplikace pro generování a aktualizaci projektů

Hlavním úkolem odborné praxe bylo vytvořit aplikaci, která vygeneruje nový projekt a spustí skript, který změní vlastnosti projektu. Aplikace byla tvořena na platformě .NET při použití WPF. V průběhu vývoje docházelo k přidávání různých dalších funkcionalit jako třeba aktualizace vlastností již vytvořeného projektu.

Časová náročnost: Tento hlavní úkol mi zabral nejvíce času, a to asi 25 dní. Největším problémem byla neznalost technologie WPF a taky přibývajících požadavků na funkčnost aplikace.

3.4 Napojení aplikace na místní IS

Posledním úkolem bylo vytvořit konzolovou aplikaci z již hotové aplikace pro generování projektu. Tuto aplikaci bylo třeba zakomponovat do již existujícího informačního systému firmy.

Časová náročnost: Napojení na místní IS trvalo přibližně 7 dní. Bylo třeba vymyslet, jak tuto aplikaci napojit, rozšířit stávající aplikaci a vše řádně otestovat.

4 Zvolený způsob řešení zadaných úkolů

4.1 Překreslení obvodových schémat

V první kapitole se budu věnovat společnosti EPLAN Software & Service. Dále také představím software této společnosti, který jsem po dobu celé odborné praxe využíval. Můj konzultant a zadavatel úkolů je přesvědčen, že nejlepší cestou pro seznámení se s novým softwarem je začít s tímto softwarem pracovat a prakticky si tak vyzkoušet jeho možnosti a funkcionality. Poslední část této kapitoly tedy věnuji úkolům, které mi byly zadány v rámci seznámení se softwarem.

4.1.1 Společnost EPLAN

Společnost EPLAN Software & Service (dále jen společnost EPLAN) vyvíjí CAE řešení a pomáhá společnostem najít způsoby, jak optimalizovat jejich inženýrské procesy. Zákazníci těží ze zvýšené efektivity při vývoji produktů prostřednictvím standardizovaných procesů, automatizovaných řešení a konzistentních pracovních postupů.

Společnost EPLAN nabízí technologickou podporu pro různé obory inženýringu. Jejich široký rozsah platforem propojuje expertní systémy z oborů:

- Mechatronika
- Elektrotechnika
- Fluidní technologie
- Měření a regulace
- Integrace

Všechny aplikace jsou vybavené stejnými základními daty a funkcemi, které podporují vysokou úroveň kvality projektů a mechatronické pracovní metody.

4.1.2 EPLAN Electric P8

EPLAN Electric P8 je software, který firma Autel používá pro realizaci veškerých projektů. Poskytuje neomezené možnosti pro projektování, dokumentaci a řízení projektů elektrotechnické automatizace. Automatická výroba se neobejde bez detailních zpráv založených na základě schémat zapojení, která jsou nedílnou součástí komplexní dokumentace a poskytují požadované údaje pro další fáze projektu, jako je výroba, montáž, uvedení do provozu a servis. Projektová data z jiných oblastí projektu mohou být vyměňována prostřednictvím rozhraní CAE, což zaručuje stabilitu a integraci v rámci celého procesu vývoje výroby.

4.1.3 Překreslení obvodových schémat

V době, kdy ještě firma Autel nevlastnila licence na aplikaci EPLAN, se projekty realizovaly na starších programech, které nejsou s dnešními moderními systémy kompatibilní. Veškerá dokumentace se tedy zachovala pouze v tištěné podobě. Firma tyto dokumentace ale pořád používá v rámci servisních prací, kde je rychlejší a pohodlnější používat jejich elektronickou verzi. Mým prvním úkolem bylo část z těchto starých dokumentací překreslit do elektronické podoby a seznámit se tak s prostředím aplikace EPLAN.

První dny byly poměrně obtížné, protože uživatelské rozhraní EPLANU je bohaté a obsahuje širokou škálu funkcí. Pro překreslení obvodových schémat jsem potřeboval pouze některé z nich. Další problém nastal při zkoumání staré papírové dokumentace. Schématické značky různých přístrojů a zařízení byly poměrně zastaralé a těžko rozeznatelné. Většinu z nich jsem ale nakonec po dlouhém zkoumání identifikoval sám nebo s pomocí pracovníka oddělení projekce. Po určení všech schématických značek, přístrojů a ovládnutí potřebných funkcí EPLANU pro kreslení obvodových schémat, jsem již neměl větší problém tento úkol dokončit.

4.2 EPLAN skripty

EPLAN, stejně jako mnoho jiných systémů či aplikací, nabízí vlastní API, které umožní vývojářům rozšiřovat aplikaci o další funkcionalitu. Toto API ale není bezplatné, a je tedy nutno si koupit licenci k jeho používání. Firma Autel tuto licenci bohužel nevlastní a bylo potřeba se spojit s bezplatnou, funkčně méně obsáhlou verzí. Ta byla součástí instalačního balíku aplikace EPLAN a poskytovala několik knihoven. Pomocí těchto knihoven bylo následně možné tvořit skripty.

V této kapitole podrobně vysvětlím, co jsou to skripty, jaké máme druhy, jak je lze vytvářet a mnoho dalších užitečných možností. Následně představím knihovny, které slouží k tvorbě skriptů. Nakonec podrobně popíšu mnou vytvořený skript. Tento skript je pak používán ke změně vlastností projektu.

4.2.1 Možnosti skriptů

V aplikaci EPLAN lze vykonávat skripty. Jsou to malé vykonatelné kódy programu, které se píší mimo aplikaci EPLAN v programovacích jazycích C# nebo Visual Basic .NET. Existují vždy jako soubory zdrojového kódu (koncovka názvu souboru *.cs nebo *.vb). V aplikaci EPLAN využívají tyto skripty rozhraní API, aniž byste k tomu potřebovali další licenci pro EPLAN API.

Skript je po spuštění načten do systému, kompilován a pak vykonán. Existují následující druhy skriptů:

- „Jednoduché“ skripty, jejichž funkce se má vykonat jen jednou, se spouští z menu přes položku nabídky **Vykonat**. Vykonají se postupně všechny akce, které skript obsahuje a ukončí se.

- Jiné skripty, které poskytují např. vlastní akce nebo vytvoření vlastní položky nabídky v menu, se pomocí položky nabídky **Načíst** trvale do aplikace EPLAN přihlásí a vykonají. Odhlášení příslušných skriptů se provede později pomocí položky nabídky **Vyjmout**.

Skriptování v aplikaci EPLAN sice neposkytuje tolik volnosti a zasahování do rozhraní jako se speciální licencí EPLAN API, ale přesto skýtá několik možností. Jednoduché skripty vykonávají akci jednorázově. Tyto skripty lze použít při automatizovaném zpracování projektu. Můžeme zde pomocí několika připravených skriptů automaticky vykonávat jednotlivé oblasti zpracování projektu (např. zálohovat data, tisk atd.). Takto lze automatizovaně zpracovat i více projektů naráz. Další využití je možnost vyčtení nebo pozměnění nastavení. Skriptováním lze také vkládat nové akce. Novou akci lze integrovat do aplikace EPLAN jako položku menu nabídky nebo tlačítko v panelu nástrojů. Můžeme také reagovat na různé události aplikace EPLAN jako např. spuštění nebo ukončení programu. Poslední možností je spuštění skriptů z příkazového řádku. Takto lze předat i vstupní parametry pro spouštěný skript.

4.2.2 Knihovny pro skripty

Ve skriptu lze používat všechny třídy montážních skupin Microsoft .NET:

- **System**
- **System.XML**
- **System.Drawing**
- **System.Windows.Forms**

a montážních skupin aplikace EPLAN API:

- **Eplan.EplApi.Base**
- **Eplan.EplApi.ApplicationFramework**

Pomocí těchto knihoven a jejich tříd lze následně vyvíjet výše zmíněné skripty. Ty se pak přeloží a zkompilují samotnou aplikaci EPLAN a spustí. K nastudování .NET knihoven společnosti Microsoft mi sloužily tyto materiály [2]. S prostudováním knihoven společnosti EPLAN už byly větší potíže. Společnost totiž neposkytuje žádnou dokumentaci, a proto bylo nutné použít anglickou dokumentaci ve formátu XML, která se dodává s danými knihovnami. I zde ovšem byly funkce popsány hodně stručně a bylo je tudíž třeba otestovat a zjistit tak jejich celkovou funkčnost.

4.2.3 Skript pro změnu vlastnosti projektu

Jak již bylo nastíněno v předchozích kapitolách, tato práce popisuje vytvoření projektu v aplikaci EPLAN s vlastnostmi zakázek, které se nacházejí v SQL databázi. Jak jsou získávány informace

z databáze, bude popsáno v následujících kapitolách. Zde se budeme věnovat změně vlastností projektu, která bude provedena pomocí výše zmíněných skriptů.

Již víme, že skripty mohou být jednoduché, provádějící pouze jednu funkci, nebo složitější, obsahující více funkcí. Já jsem použil jednoduchý skript, který obsahoval jednu funkci se vstupními parametry. Tato funkce obsahovala spouštění specifické akce EPLANU, která změnila vlastnost projektu. Tato akce byla spouštěna tolikrát, kolik vlastností bylo potřeba změnit.

Tabulka 1: Akce XEsSetProjectPropertyAction

Parametr	Popis
PropertyId	Identifikátor stanovované vlastnosti (číslo vlastnosti)
PropertyIndex	Má-li vlastnost indexy, jde o index; většinou 0
PropertyValue	Nová hodnota vlastnosti

Následující část kódu napsaná v jazyce C# je ukázkou použití akce XEsSetProjectPropertyAction. Jedná se pouze o část programu, nikoli kompletní řešení.

```
ActionCallingContext action = new ActionCallingContext();
CommandLineInterpreter cm = new CommandLineInterpreter();
action.AddParameter("PropertyId", "10011");
action.AddParameter("PropertyIndex", "0");
action.AddParameter("PropertyValue", Param1);
cm.Execute("XEsSetProjectPropertyAction", action);
```

Výpis 1: Použití akce XEsSetProjectPropertyAction

Tento jednoduchý skript obsahuje funkci se vstupními parametry. Je tedy potřeba spouštět skript z příkazové řádky. K tomuto spuštění jsem potřeboval využít další akce EPLANU, které je možno spustit z příkazového řádku. První z těchto akcí je *ProjectAction*. Tato akce má dva povinné parametry. Jedním je *Project*, který požaduje kompletní cestu k projektu, na kterém chceme provést danou akci, a druhým je akce samotná. Druhá akce je *ExecuteScript*, která spustí můj skript na změnu vlastností. Cestu ke skriptu uvedeme za parametrem *ScriptFile*. Nakonec už následuje výčet parametrů, které očekává spouštěný skript. Celý tento příkaz začíná parametrem */C*, který umožní tento příkaz spustit a následně ukončí aplikaci příkazového řádku. Poslední věc, kterou příkaz obsahuje, je absolutní cesta ke spouštěcímu souboru aplikace EPLAN s několika parametry. Ty umožňují, že se EPLAN spustí na pozadí v tichém módu a po provedení skriptu se ukončí.

```
/C "./EPLAN.exe" /NoLoadWorkspace /NoSplash /Frame:0 /Auto /Quiet ProjectAction
/Project: "./Projekt.elk" /Action:ExecuteScript /ScriptFile "./
SetProjectProperty.cs" /Param1:"prvni parameter" /Param2:"druhy parameter"
...
```

Výpis 2: Ukázka spuštění skriptu v příkazovém řádku

4.3 WPF aplikace pro generování a aktualizaci projektů

V této kapitole důkladně popíšu aplikaci, která slouží pro generování a aktualizaci projektů aplikace EPLAN. Nejprve stručně popíšu technologii WPF, kterou jsem si vybral pro tvorbu grafického rozhraní. Dále vysvětlím postup vývoje aplikace a metodiky, které jsem při tom použil. Stručně taky představím způsob připojení k databázi a práci s daty. Nakonec znázorním a popíšu průběh generování a aktualizaci projektu a také představím grafické rozhraní aplikace a jeho možnosti. Tato aplikace byla vyvíjena na platformě .NET Framework v programovacím jazyce C#.

4.3.1 WPF

Windows Presentation Foundation [4] (zkratkou WPF) je podmnožina .NET Frameworku od verze 3.0, který používá značkovací jazyk XAML pro vytvoření „uživatelsky bohatého rozhraní“. Cílem je tedy sjednotit uživatelské rozhraní, 2D a 3D grafiku, vektorovou a rastrovou grafiku, animace, vázání dat a audio a video. K tomu nám poslouží značkovací jazyk XAML, který odděluje funkčnost a vzhled aplikace. XAML je založen na XML a je tedy velmi jednoduché se v něm vyznat a pracovat s ním.

Tuto technologii jsem se rozhodl vybrat pro prezentační část mé aplikace. Již v minulosti jsem pracoval s podobnou technologií společnosti Microsoft Windows Forms (zkráceně WinForms). S WinForms už se ale v dnešní době moc nepracuje a používá se modernější a rozšířenější WPF. K seznámení a pochopení technologie WPF mi dobře posloužil internetový tutoriál a stránky společnosti Microsoft [5] [6].

4.3.2 Databáze

Jak již bylo naznačeno, jednou z částí mého úkolu bylo získat určitá data z databáze pro další zpracování. Firma Autel uchovává všechna data na lokálním serveru v podobě SQL databáze. Tato databáze obsahuje asi sto tabulek. Moje aplikace ovšem využívala pouze pár z nich. Při vývoji mé aplikace jsem pracoval pouze s kopií těchto pár tabulek s testovacími daty. Po dokončení a schválení mé aplikace jsem se pak mohl připojit přímo na lokální databázi.

Pro namapování relačních dat z databáze na objekty jsem použil návrhový vzor Data Mapper [7]. Tento vzor jsem již dříve používal v rámci studia a měl jsem tedy vyzkoušenou jeho implementaci. Princip tohoto návrhového vzoru tkví v oddělení mapovacího objektu od doménového. Mapovací objekt se stará o CRUD operace a komunikuje s databází na rozdíl od doménového objektu, který je zcela nezávislý na použité databázi. Výhodou tohoto vzoru je právě nezávislost doménového objektu na datovém modelu.

Při prezentaci této aplikace vedoucímu IT oddělení, mi byl přístup k databázovým objektům povolen a celé objektově-relační mapování schváleno. Pro příští použití mi bylo doporučeno využití Entity Frameworku, který práci objektově-relačního mapování výrazně usnadňuje.

4.3.3 Vývoj aplikace

Tuto aplikaci jsem samostatně navrhoval, implementoval a testoval, a proto mi zabrala podstatnou část mého působení ve firmě. Aplikace byla v průběhu stále rozšiřována o další funkcionalitu, a proto se její vývoj dá přirovnat k inkrementálnímu přístupu. Opakovaně jsem tedy prováděl sekvenci malých úkolů až do doby, kdy byl můj konzultant s aplikací spokojen a nebylo tedy třeba dále vyvíjet. Tato sekvence vypadala následovně:

1. **Požadavky.** Konzultant mi sdělil své požadavky a já jsem se případně doptal na různé detaily.
2. **Návrh.** Z přidělených požadavků jsem zpracoval návrh, který jsem krátce představil konzultantovi.
3. **Implementace.** Podle návrhu jsem implementoval danou část.
4. **Kontrola a test.** Po implementaci jsem danou část otestoval a představil konzultantovi. Ten zkontroloval, zda jsou funkční jednotlivé požadavky a případně navrhl změnu.

Jelikož můj konzultant nepracuje v oboru IT, byla některá rozhodnutí ponechána na mně. On si pak pouze ověřil funkčnost zadaného požadavku.

Z důvodu přehlednosti a lepší rozšiřitelnosti jsem se rozhodl aplikaci implementovat pomocí třívrstvé architektury. Rozdělení je následující:

- **Datová vrstva** - úkolem této vrstvy je poskytnout data vrstvě vyšší. Bylo potřeba implementovat již zmíněný návrhový vzor Data Mapper. Vytvořil jsem tedy mapovací třídy, které získaly data z databáze pomocí jednoduchých CRUD operací a poslaly je vyšší vrstvě v datové struktuře `DataTable`.
- **Aplikační vrstva** - tato vrstva obsluhuje kompletní logiku aplikace, provádí výpočty a zpracovává data. Přijímám zde data z nižší datové vrstvy a ukládám je do objektů.
- **Prezentační vrstva** - tato vrstva slouží pro interakci s uživatelem pomocí grafického uživatelského rozhraní. Je tvořena technologií WPF. Má přístup ke všem funkcím aplikační vrstvy.

4.3.4 Generování projektu

V této sekci podrobně popíšu celý proces generování projektů pro aplikaci EPLAN. Počátečním problémem bylo vymyslet, jak vytvořit nový projekt bez použití grafického rozhraní EPLANU. Jelikož skripty v aplikaci nenabízely žádné řešení pro vytvoření nového projektu, bylo potřeba vymyslet jinou cestu. Po delším uvažování jsem vymyslel řešení, které mi bylo následně schváleno i ze strany konzultanta.

Dekomprimace šablony Toto řešení zahrnovalo vytvoření šablony projektu v EPLANU. Ta byla zkomprimována do ZIP formátu a uložena na určené místo na serveru. První částí generování bylo tedy dekomprimovat šablonu projektu na místo určené pro daný projekt. Před tímto procesem bylo potřeba zkontrolovat nebo sestavit správné cesty souborů. Jako první se kontrolovala cesta k ZIP souboru se šablonou. Při nenalezení se celý proces generování ukončil s varovnou hláškou. Druhá na řadě byla cílová cesta uložení projektu. Tuto cestu bylo potřeba sestavit podle čísla vybrané zakázky, která v sobě obsahovala rok. Díky tomu jsem mohl vybrat složku se správným rokem. Poslední částí bylo vytvoření složky pro daný projekt, která byla pojmenována podle čísla a názvu zakázky. Pokud již daný projekt existoval, proces byl ukončen s chybovou hláškou.

Vytvoření vlastností Další fází procesu bylo vytvoření objektu, který obsahoval veškeré vlastnosti dané zakázky. V této fázi bylo potřeba několikrát přistoupit do databáze, abychom získali všechna potřebná data. Většina vlastností byla připravena a nebylo třeba je před další fází nijak upravovat. Problém ovšem nastal s aktuálním datem. EPLAN uchovává datum a čas v Unixovém časovém formátu [8]. Tento čas je vyjadřován v podobě sekund, které uběhly od Unixové epochy, která je stanovena na 1.1. 1970 00:00:00 až po dané datum a čas. Jelikož .NET Framework neposkytuje žádnou funkci pro tuto konverzi, bylo potřeba si vytvořit funkci vlastní.

```
public static double ConvertToUnixTimestamp(DateTime date)
{
    DateTime origin = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc);
    TimeSpan diff = date.ToUniversalTime() - origin;
    return Math.Floor(diff.TotalSeconds);
}
```

Výpis 3: Funkce pro převedení na Unixový čas

Tato funkce obsahuje jako vstupní parametr `DateTime` proměnnou, která představuje datum a čas, které chceme převést. Od vstupní časové proměnné odečteme datum, od kterého se počítá Unixový čas. Funkce pak vrátí celkový počet sekund tohoto časového rozdílu.

Spuštění skriptu Po úspěšném dekomprimování projektu a vytvoření vlastností zbývá pouze spustit skript pro změnu vlastností projektu. Tento skript je detailněji popsán v kapitole 4.2. Před spuštěním tohoto skriptu bylo zapotřebí zkontrolovat několik cest. První byla cesta ke skriptu a druhá ke spustitelnému souboru aplikace EPLAN. Obě tyto cesty musely být validní a existovat, jinak se celý proces ukončil s chybovou hláškou. Pokud byly obě cesty v pořádku, skript se spustil a změnil tak vlastnosti projektu. Tímto celý proces úspěšně skončil.

4.3.5 Aktualizace projektů

Výše popsaný proces generování projektu slouží k vytvoření zcela nového projektu. Může ale nastat situace, kdy je již projekt vytvořen a vyvíjen a je potřeba v něm pouze změnit vlastnosti, tedy aktualizovat projekt.

K aktualizaci projektu byl částečně použit proces pro generování projektu. Jelikož už ale projekt existuje, byla vynechána část, která projekt dekomprimuje ze šablony do cílové složky. Tato část byla nahrazena zobrazením souborového dialogu, ve kterém bylo třeba vybrat již existující projekt, který chceme aktualizovat. Následně již byly použity hotové části pro vytvoření vlastností a spuštění skriptu.

4.3.6 Zpracování loga firmy

Při tvorbě obvodových schémat v aplikaci EPLAN se jednotlivé stránky zobrazují jako výkresy. Každý z těchto výkresů má razítko, které obsahuje jednotlivé informace o daném výkresu a osobě, která tento výkres vytvořila. Dále také obsahuje informace o firmách, jak zhotovitele, tak objednatele či koncového zákazníka. K těmto informacím jsou také přidána loga jednotlivých firem. Mým úkolem bylo zakomponovat práci s těmito logy do aktuální aplikace.

Jako úložiště pro jednotlivá firemní loga jsem zvolil samostatnou složku na serveru, na kterém jsou ukládány i jednotlivé projekty. Logo lze do projektu zakomponovat jeho zkopírováním do vnořené složky, která obsahuje externí data projektu. V EPLANU se pak pouze určí jako zdroj pro loga právě tato složka.

Celý proces pro práci s logy je zakomponovaný v procesu generování i aktualizaci projektu. Pokud chceme logo k dané firmě přidat, musíme jej vybrat ještě před provedením akce generování nebo aktualizace pomocí dialogového okna. Pokud se logo pro danou firmu již nachází ve složce na serveru, je vybráno toto logo. Po následném provedení akce generování či aktualizace se nejprve přkopíruje logo do složky pro externí data projektu. Následně se zkontroluje, jestli je logo obsažené ve složce pro loga firem. Pokud ne, logo se zde zkopíruje. Tímto celý proces pro práci s logy firem skončí.

Tento proces je používán dvakrát za sebou, protože pracujeme s logy pro objednatele a koncového zákazníka.

4.4 Napojení aplikace na místní IS

Posledním zadaným úkolem bylo napojit aplikaci pro generování a aktualizaci projektu na místní informační systém. Součástí bylo prostudovat místní systém a vymyslet řešení, jak by bylo možné dané propojení realizovat. V této kapitole tedy stručně popíšu místní informační systém a následně realizaci napojení.

4.4.1 Firemní informační systém

Jedná se o malý ERP systém zaměřený na inženýringovou firmu, která se zabývá realizací projektů. Základ tohoto systému tvoří databáze SQL Server od společnosti Microsoft. Systém se skládá ze dvou uživatelských rozhraní. Jedno slouží pro vedení, management, marketing a obchodníky, které se stará o vlastní práci s projekty. Je založeno na Javě. Hlavní funkcí tohoto rozhraní je evidence nabídek a projektů. Umožňuje také sledovat vytíženost zaměstnanců či oddělení. Lze přes něj rovněž přidělovat práci na projektech jednotlivým zaměstnancům. Druhé rozhraní je dostupné pro všechny zaměstnance. Jedná se o webové rozhraní vyvíjené technologií ASP.NET. Jednou ze základních funkcí tohoto rozhraní je nástroj pro plánování úkolů. Pomocí tohoto nástroje si zaměstnanec rozepisuje jemu přidělenou práci do jednotlivých úkolů. Lze takto efektivně přidělovat zdroje pro jednotlivé projekty a hlídat při tom jeho stav. K tomuto systému je pak ještě připojen účetnický systém.

4.4.2 Napojení na IS

Ke správnému vygenerování projektu jsou potřebné dvě hlavní věci. První je číslo projektu, podle kterého identifikujeme daný projekt. Druhou je aplikace EPLAN bez které bychom projekt nevygenerovali. Aplikace, kterou jsem vytvořil, potřebuje ke svému používání interakci s uživatelem. Pokud chceme aplikaci napojit na informační systém, bude potřeba ji rozšířit tak, aby byla schopna vygenerovat projekt samostatně při znalosti čísla projektu. Druhé kritérium vyřešíme umístěním aplikace na server, na kterém běží systém Windows a je zde obsažena aplikace EPLAN.

Rozšíření aplikace Aplikaci jsem rozšířil o možnost spouštění z příkazového řádku, ve které můžu zadat argument. Tímto argumentem pak bude potřebné číslo projektu. Aplikace tak nebude potřebovat obsluhu uživatele. Konkrétní realizace proběhla přepsáním metody `OnStartup`, která se provádí při spuštění aplikace. Pokud nejsou zadány žádné argumenty, aplikace se spustí v desktopovém režimu s uživatelským prostředím. V případě spuštění s argumentem se počítá, že jde o číslo projektu, a provede se proces pro generování projektu. Při výskytu chyby se hláška vypíše do konzole. Všechny cesty, které jsou potřebné v procesu generování, musí být zadány v konfiguračním souboru aplikace. Pokud jsou tyto cesty nastaveny správně, pak generování proběhne úspěšně. Takto lze vygenerovat i víc projektů najednou a to zadáním několika čísel projektů oddělených středníkem.

Následující body popisují celkový průběh procesu generování informačním systémem:

1. Do informačního systému je vložena zakázka pro realizaci projektu.
2. Spustí se „spouštěč“ (anglický trigger), který vloží do tabulky úkolů, že se má provést vygenerování tohoto projektu. Tabulka úkolů je zpracovávána jiným procesem a samotné generování proběhne až později.

3. Proces, který provádí jednotlivé úkoly, narazí na úkol pro vygenerování projektu a provede ho. Toto je realizováno spuštěním dávkového souboru na serveru, kde se aplikace pro generování nachází.
4. Dávkový soubor pak spustí mojí aplikaci, která provede vygenerování projektu.

Implementace týkající se databáze byla provedena pracovníkem IT oddělení z důvodu znalosti struktury a fungování vlastní databáze.

5 Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe

V průběhu odborné praxe jsem využil mnoho znalosti získaných při studiích na vysoké škole. V první řadě se jednalo o předměty Úvod do softwarového inženýrství a Vývoj informačních systému. Z těchto předmětů jsem využil teoretické znalosti o vývoji softwaru a také použití vícevrstvé architektury aplikace spolu s implementací návrhového vzoru. Pracoval jsem taky s databází SQL, kde jsem zúročil znalosti z předmětu Úvod do databázových systému a Databázové a informační systémy. Co se týká samotného programování a použitých programovacích jazyků, využíval jsem pouze technologie .NET Frameworku společně s jazykem C#. Tyto znalosti jsem nabyl absolvováním předmětů Programovací jazyky 2 a hlavně Architektura technologie .NET. V neposlední řadě stojí za zmínku předmět Elektronické publikování, díky kterému jsem mohl využít znalosti L^AT_EXu při tvorbě této práce.

6 Znalosti či dovednosti scházející studentovi v průběhu odborné praxe

V průběhu odborné praxe jsem samostatně vyvíjel software podle zadaných požadavků. Chyběla mi zde ovšem větší zkušenost a vývoj se tedy neobešel bez chyb, které vznikaly třeba ze špatného pochopení požadavku. Vzhledem k mému studijnímu oboru byly mé největší nedostatky při plnění prvního úkolu, kdy jsem pracoval s profesionálním CAD/CAE nástrojem EPLAN a kreslil v něm obvodová schémata. Další scházející znalostí byla technologie WPF, ve které jsem tvořil uživatelské rozhraní aplikace.

7 Dosažené výsledky v průběhu odborné praxe a její zhodnocení

Díky absolvování odborné praxe jsem si rozšířil vědomosti o programovacím jazyce C#. S tím souvisí seznámení s technologií WPF, se kterou jsem se ve škole nesetkal. Další velkou zkušeností byla práce v profesionálním CAD/CAE nástroji EPLAN, ve kterém jsem kreslil obvodová schémata. Díky tomu jsem měl možnost nahlédnout a vyzkoušet si zcela jinou část vývoje produktu, a to z hardwarové části. Měl jsem také možnost nahlédnout na spolupráci třech oddělení při sestavování a testování výsledného produktu. Konkrétně montérů, kteří produkt sestavovali, IT odborníků, kteří pracovali na softwaru a pak projektantů, kteří celou práci řídili a testovali. Viděl jsem tak různé fáze vzniku produktu, jak z hardwarové stránky, tak softwarové. Možnost spolupracovat ve větším týmu jsem sice neměl, ale zkušenosti, které jsem získal při samostatném vypracovávání úkolů, byly pro mě velkým přínosem.

S vykonáním odborné praxe ve firmě Autel jsem velice spokojen. Díky práci na reálných projektech s budoucím využitím jsem získal mnoho znalostí a zkušeností.

Literatura

- [1] O společnosti Autel a.s.
Autel [online]. © 2015 [cit. 2016-03-26].
Dostupné z: <http://www.autel.cz/o-spolecnosti/profil-spolecnosti/>
- [2] Class Library.
MSDN [online]. © [cit. 2016-04-01].
Dostupné z: [https://msdn.microsoft.com/cs-cz/library/d11h6832\(v=vs.71\).aspx](https://msdn.microsoft.com/cs-cz/library/d11h6832(v=vs.71).aspx)
- [3] EPLAN Electric P8.
EPLAN [online]. © 2016 [cit. 2016-04-03].
Dostupné z: <http://www.eplan.cz/cz/reseni/elektrotechnika/eplan-electric-p8/>
- [4] Windows Presentation Foundation.
Wikipedie [online]. © 2015 [cit. 2016-04-07].
Dostupné z: https://cs.wikipedia.org/wiki/Windows_Presentation_Foundation
- [5] Windows Presentation Foundation tutorial.
wpf-tutorial [online]. © 2012-2016 [cit. 2016-04-07].
Dostupné z: <http://www.wpf-tutorial.com/>
- [6] Introduction to WPF.
MSDN [online]. © [cit. 2016-04-07].
Dostupné z: [https://msdn.microsoft.com/cs-cz/library/aa970268\(v=vs.100\).aspx](https://msdn.microsoft.com/cs-cz/library/aa970268(v=vs.100).aspx)
- [7] Data Mapper.
Wikipedie [online]. © 2013 [cit. 2016-04-07].
Dostupné z: https://cs.wikipedia.org/wiki/Data_Mapper
- [8] Unix time.
Wikipedie [online]. © 2014 [cit. 2016-04-08].
Dostupné z: https://en.wikipedia.org/wiki/Unix_time

8 Přílohy

I. Příloha na CD

Elektronická příloha na disku CD obsahuje složku SQL_klient. V této složce jsou pak zdrojové kódy výsledné aplikace.